(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 514 043 A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 92303900.2

(22) Date of filing: 30.04.92

(51) Int. Cl.5: G06F 15/16

(30) Priority: 13.05.91 US 698866

(43) Date of publication of application: 19.11.92 Bulletin 92/47

(84) Designated Contracting States: DE FR GB IT

(71) Applicant: International Business Machines Corporation
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: Rolfe, David Bruce
24 Pine Tree Road
West Hurley, New York 12491(US)

(74) Representative: Blakemore, Frederick Norman et al
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) Computing system.

(57) Computer elements in a massively parallel computer system are interconnected in such a way that the number of connections per element can be balanced against the network diameter or worst case path length. This is done by creating a topology that maintains topological properties of hypercubes yet improves flexibility by enumerating the nodes of the network in number systems whose base can be varied. Topologies are generated in which nodes are not always connected when their addresses differ in a single digit. A new variable d is introduced, the purpose of which is to control the overall density of the network by controlling the number of intermediate arc connections within the rings of the network.

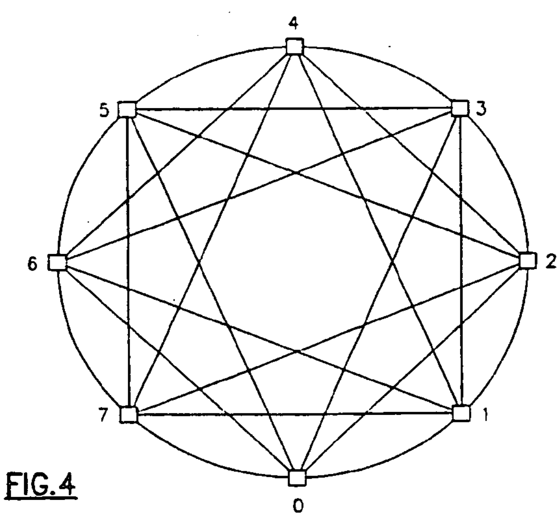

FIG.4

EP 0 514 043 A2

The present invention relates to massively parallel computing systems.

As the cost of computer elements declines, the possibility of building very large networks is being realised. Ralph Duncan in "A Survey of Parallel Computer Architectures", Computer, February 1990, pp. 5 to 16, reviews alternative approaches to parallel processing. One class of parallel architecture described by Duncan involves multiple processors autonomously executing diverse instructions on diverse data, which Duncan classifies as MIMD for multiple instruction, multiple data streams. Within this class of parallel architecture are the ring, mesh, tree, and hypercube topologies. The ring topology is characterised as having a communication diameter of N/2 where N is the number of nodes. The communication diameter is defined as the worst case path length. In contrast, a two-dimensional mesh topology typically has a communication diameter of 2(N-1) where, again, N is the number of nodes, and this is also the typical communication diameter of tree topologies, although in both cases strategies have been employed to reduce the communication diameters.

Of these known network topologies, the hypercube has some interesting topological properties. A Boolean n-cube or "hypercube" topology uses $N=2^n$ processors arranged in an n-dimensional array. Each node, or processor, in the array has $n=\log_2 N$ bidirectional links to adjacent nodes. Figure 1 illustrates hypercubes for n=2,3,4. Examples of existing hypercube machines include Caltech's CosmicCube, Intel's iPSC/2 and NCube's Ncube/10. Descriptions of these and other hypercube computer systems are provided by John P. Hayes et al. in "Hypercube Supercomputers", Proceedings of the IEEE, vol. 77, no. 12, December 1989, pp. 1829 to 1841. Ophir Frieder in "Multiprocessor Algorithms for Relational-Database Operations on Hypercube Systems", Computer, November 1990, pp. 13 to 28, provides a tutorial on hypercube systems, particularly as database engines. Hayes et al. list a variety of representative applications for hypercube computers.

With specific reference to Figure 1, note the numbering convention for the addresses of the nodes. For n=2, there are four nodes ($N=2^2$) with addresses 00, 01, 01, and 11. Each node is connected to two other nodes, and the rule applied is that each node is connected to an adjacent node having an address which differs by only one bit. For n=3, there are eight nodes ($N=2^3$) with addresses 000, 001, 010, 011, 100, 101, 110, and 111, but the connection scheme is the same with, for example, node 000 being connected to nodes 001, 010 and 100. There are sixteen nodes for the case where n=4 ($N=2^4$), but still the same connection scheme is employed. Laxmi N. Bhuyan and Dharma P. Agrawal in "Generalized Hypercube and Hyperbus Structures for a Computer Network", IEEE Transactions on Computers, vol. c-33, no. 4, April 1984, pp. 323-333, describe a generalisation of the binary hypercube to a variable radix numbering. In this scheme, the nodes of a network are numbered in a system in which each digit of the number can be based on a different radix. Nodes are connected whenever the resulting node addresses differ in exactly one digit.

While hypercube architectures offer some very attractive topological properties, as these networks go beyond a few thousand nodes, the numbers of connections per node and path lengths grow in fixed ways. If either connections per node or path lengths become too large for a given technology, the designer is forced to develop new network algorithms from scratch. These new ad hoc topologies may then adversely effect the performance of existing programs that ran well on smaller networks with older topologies.

According to the present invention, there is provided a computing system in the form of a non-binary hypercube comprising:

a plurality of nodes, N, where $N=b^n$, b and n being positive integers and b>2, each node having an identifying address numbered according to a number system base b; and

means for interconnecting each node to nodes having addresses differing from the node's address by exactly one digit, the number of connections per node being $\mu \log_b N$, where $\mu=2d$, $1 \leqq d \leqq b/2$ if b is even and $1 \leqq d \leqq (b-1)/2$ if b is odd, except if b is even and $d=b/2$, in which case $\mu=b-1$.

The present invention enables a method of interconnecting processors to form a generalised hypercube structure to be provided that supports an unlimited number of nodes.

In such a computer structure, the number of interconnections per processor can be adjusted against the network diameter. According to the invention, there is provided a method for interconnecting processing elements in such a way that the number of connections per element can be balanced against the network diameter. This is done by creating a topology that maintains many of the well known and desirable topological properties of hypercubes while improving flexibility by enumerating the nodes of the network in number systems whose base can be varied. When using a base two number system, this method creates the familiar Boolean binary hypercube topology. However, in contrast to a Boolean binary hypercube, the practice of the invention results in a non-binary hypercube having fewer interconnections permitting the practical realisation of very large computer systems having an unlimited number of nodes.

The subject invention is a refinement of the scheme described by Laxmi N. Bhuyan and Dharma P. Agrawal, supra. The invention generates topologies in which nodes are not always connected when their addresses differ in a single digit. The invention introduces a new variable d, the purpose of which is to control the overall density of the network by controlling the number of intermediate arc connections within the rings of the network. For a network having N nodes, the assumption is made that there exist positive integers b and n such that $N = b^n$. The new variable d is picked such that $1 \leq d \leq b/2$, if b is even, and $1 \leq d \leq (b-1)/2$, if b is odd. The nodes of the network are numbered to the base b. Two nodes, x and y, of the network if and only if

1. the address of x differs from the address of y in exactly one digit, and
2. for the digit i where x_i is not equal to y_i, $y_i = (x_i + j) \bmod b$ or $y_i = (x_i - j) \bmod b$ for some $j \leq d$.

The invention will be better understood from the following detailed description of embodiments of the invention with reference to the drawings, in which:

Figure 1 is a pictorial representation of some exemplary Boolean hypercube topologies;
Figure 2 shows the topology of a typical ring in a base 8 network where d = 1;
Figure 3 shows the topology of a typical ring in a base 8 network where d = 2;
Figure 4 shows the topology of a typical ring in a base 8 network where d = 3;
Figure 5 shows the topology of a typical ring in a base 8 network where d = 4;
Figure 6 is a block diagram of a 125 computer network where b = 5, n = 3 and d = 1;
Figure 7 is a block diagram of the lowest level of packaging of an implementation of the invention composed of 4096 nodes;
Figure 8 is a block diagram of the next level of packaging of the embodiment shown in Figure 7; and
Figure 9 is a block diagram of the highest level of packaging of the embodiment shown in Figure 7 of the drawings.

Before describing a specific implementation of the invention, the network connection algorithm is discussed in general terms. The algorithm is based on the assumption that there exists positive integers b and n such that $N = b^n$, where N is the number of nodes in the network. An integer d is chosen between one and b/2, if b is even, or between one and (b-1)/2, if b is odd. Once the integer d has been selected, the nodes of the network are numbered to the base b. In other words,

$$N = a_m b^m + a_{m-1} b^{m-1} + \ldots + a_1 b^1 + a_0 b^0,$$

where $a_0, a_1, \ldots, a_{m-1}, a_m$ are non-negative integers each less than b. For a base ten (b = 10) number system, the symbols a_i are called "digits", and for a base two (b = 2) number system, the symbols a_i are called "bits". Without loss of generality, the symbols a_i are referred to hereinafter as "digits" even though the base number system may be other than ten.

Next, connect two nodes, x and y, of the network if and only if

1. the address of x differs from the address of y in exactly one digit, and
2. for the digit i where x_i is not equal to y_i, $y_i = (x_i + j) \bmod b$ or $y_i = (x_i - j) \bmod b$ for some $j \leq d$.

Notice that -x mod b = (y-x) mod b. For example, if b = 5 and d = 1, then a node whose address is 1234 would be connected to nodes having addresses 2234, 0234, 1334, 1134, 1224, 1244, 1233, and 1230. Notice that d controls the density of the network. The larger d becomes, the denser the network becomes.

During the application of the algorithm, the ports of the various nodes are labelled to indicate both the digit and digit value of the node to which they connect. For example, a port labelled 3,7 would connect to a node whose third digit is 7. The number of connections per node is $\mu \log_b N$, where $\mu = 2d$, except if b is even and d = b/2 in which case $\mu = b-1$. This can be seen to be true by considering that for each integer less than or equal to d, there are two connections for each digit of a node address. The number of digits in the node address is $\log_b$. The diameter of the network is given by

$$\frac{b-1}{\mu}(\log_b N),$$

where

$$\frac{b-1}{\mu}$$

is always rounded up to the next integer when it is not an integer.

Referring now to the drawings, and more particularly to Figure 2, there is shown the topology of a typical ring in a base 8 (i.e., b = 8) network where d = 1. There are eight computers or nodes, N, in this network so that n = 1 or $N = 8^1$. While this illustration is trivial, the next three figures illustrate the effect of increasing the value of d on the density of the network. Figure 3 shows the case for

d = 2. In this case, each computer connects to four other computers. Figure 4 shows the case for d = 3. In this case, each computer connects to six other computers. Figure 5 shows the case for d = 4. In this case each computer connects to seven other computers. Thus, Figures 2, 3, 4, and 5 provide a graphic illustration of the increasing density of a simple eight node network for an increasing value of d.

A more practical application of the interconnection technique according to the invention is illustrated in Figure 6 which shows a 125 computer network. In this example, the base, b, is set to equal five so that n = 3; that is, $N = b^n = 5^3 = 125$. In order to minimise the density of this network, d is set equal to one. This network is implemented in five planes 10, 11, 12, 13, and 14, with each plane having twenty-five computers 15. Within each plane there are five rings and five loops. Each computer in the network is connected to six other computers, two computers within its ring, two computers within its loop and two computers in different planes.

Even for this still relatively simple network, the selection of the variable d allows a control of the density of the network not known in the prior art. The network shown in Figure 6 may be easily realised as a practical embodiment of five printed circuit boards with edge connectors. However, the invention is especially useful for very large networks where interconnection densities become critical to whether an implementation is realisable in the real world.

To illustrate this, a specific implementation of the invention for d = 1, b = 8 and n = 4 is shown in Figures 7, 8 and 9. Thus, for this implementation $N = b^n = 8^4 = 4096$. For this very large number of nodes, the density variable d is chosen to be one to minimise the density.

Figure 7 shows the lowest level of packaging. Each box 20_0 to 20_7 contains eight computers interconnected with forty-eight (3x16) data paths 22_1, 22_2 and 22_3 per box. The complete package 24 in Figure 7 thus contains sixty-four (8x8) computers. The package 24 is interconnected to the rest of the computers in the system with 256 (2x128) data paths 26_1 and 26_2.

Figure 8 shows the next level of packaging. Each package 24_0 to 24_7 contains 64 computers with 256 data paths as shown in Figure 7. The larger package 30 thus contains 512 (8x64) computers supporting 1024 (2x512) data paths. Figure 9 shows the highest level of packaging containing eight packages 30_0 to 30_7, each containing 512 computers and 1024 data paths. There are 4096 (8x512) computers with 1024 data paths in highest level of packaging.

In this network of 4096 nodes, each node is directly connected only eight other nodes. This permits any node to communicate with any other node in the network by involving no more than sixteen intermediate nodes. This compares well with the case of the Boolean binary hypercube which demands a denser network with each node connected to twelve other nodes rather than eight. Thus, the invention allows for a reduction of 33% in wiring density as compared with the Boolean binary hypercube. This reduction, however, is traded for a concomitant 33% increase in network diameter.

To route messages in a network according to the invention, first compare the destination address with the current address. If they are the same, routing has been completed. If not, select some digit, i, for which the current and destination addresses are different. Forward the message out of the port for this digit that differs from the destination address by the least amount. Continue this procedure until the current and destination addresses are equal. Note that the selection of the digit to work on at a particular time can be based on which ports are not busy or damaged. This is common procedure for hypercubes. A port is said to be damaged if the port itself is damaged or the port to which it connects is damaged.

The invention simplifies the interconnection of massively parallel computer systems yet preserves the desirable properties of a hypercube. These properties are (1) a large number of alternate paths, (2) very high aggregate bandwidth, and (3) uniform connections. Existing methods can be used to used to route messages within the network. Generally, the invention provides a non-binary hypercube with less density thereby allowing the practical realisation of massively parallel computer systems having an unlimited number of nodes, but as illustrated in Figure 6, the invention also has practical application for smaller computer networks.

Computer elements in a massively parallel computer system are interconnected in such a way that the number of connections per element can be balanced against the network diameter or worst case path length. This is done by creating a topology that maintains topological properties of hypercubes yet improves flexibility by enumerating the nodes of the network in number systems whose base can be varied. Topologies are generated in which nodes are not always connected when their addresses differ in a single digit. A new variable d is introduced, the purpose of which is to control the overall density of the network by controlling the number of intermediate arc connections within the rings of the network.

While the invention has been described in terms of illustrative and preferred embodiments, those skilled in the art will recognise that the invention can be practised with modification within the

scope of the appended claims.

Claims

1. A computing system topologically arranged to form a non-binary hypercube comprising:
   a plurality of nodes, N, where $N = b^n$, b and n being positive integers and b>2, each node having an identifying address numbered according to a number system base b; and
   means for interconnecting the nodes so that each node is connected to nodes having addresses differing from the node's address at exactly one digit, the number of connections per node being $\mu \log_b N$, where $\mu = 2d$, $1 \leq d \leq b/2$ if b is even and $1 \leq d \leq (b-1)/2$ if b is odd, except if b is even and $d = b/2$, in which case $\mu = b-1$.

2. A computing system as claimed in Claim 1 wherein two nodes, x and y, of the network are connected if and only if
   1. the address of x differs from the address of y in exactly one digit, and
   2. for the digit i where x_i is not equal to y_i, $y_i = (x_i + j) \bmod b$ or $y_i = (x_i - j) \bmod b$ for some $j \leq d$.

3. A computing system as claimed in Claim 1 or Claim 2 wherein b = 8, n = 1, and d = 2.

4. A computing system as claimed in Claim 1 or Claim 2 wherein b = 8, n = 1, and d = 3.

5. A computing system as claimed in Claim 1 or Claim 2 wherein b = 8, n = 1, and d = 4.

6. A computing system as claimed in Claim 1 or Claim 2 wherein b = 5, n = 3, and d = 1.

7. A computing system as claimed in Claim 1 or Claim 2 wherein b = 8, n = 4, and d = 1.

8. A method of interconnecting a plurality of nodes in a network comprising the steps of:
   defining b and n such that the number N of nodes equals b^n and b and n are positive integers and b>2;
   choosing a positive integer d such that $1 \leq d \leq b/2$ if b is even and $1 \leq d \leq (b-1)/2$ if b is odd;
   addressing each of the nodes by numbering them using numbers represented by a base b numbering system; and
   connecting two nodes if and only if the address of a first one of said two computing elements differs from the address of the second one of said two nodes in exactly one digit, the i^{th} digit and, for the unique i^{th} digit of the addresses of said two computing elements, symbolised by x_i and y_i, respectively, where x_i is not equal to y_i, $y_i = (x_i + j) \bmod b$ or $y_i = (x_i - j) \bmod b$ for some $j \leq d$.

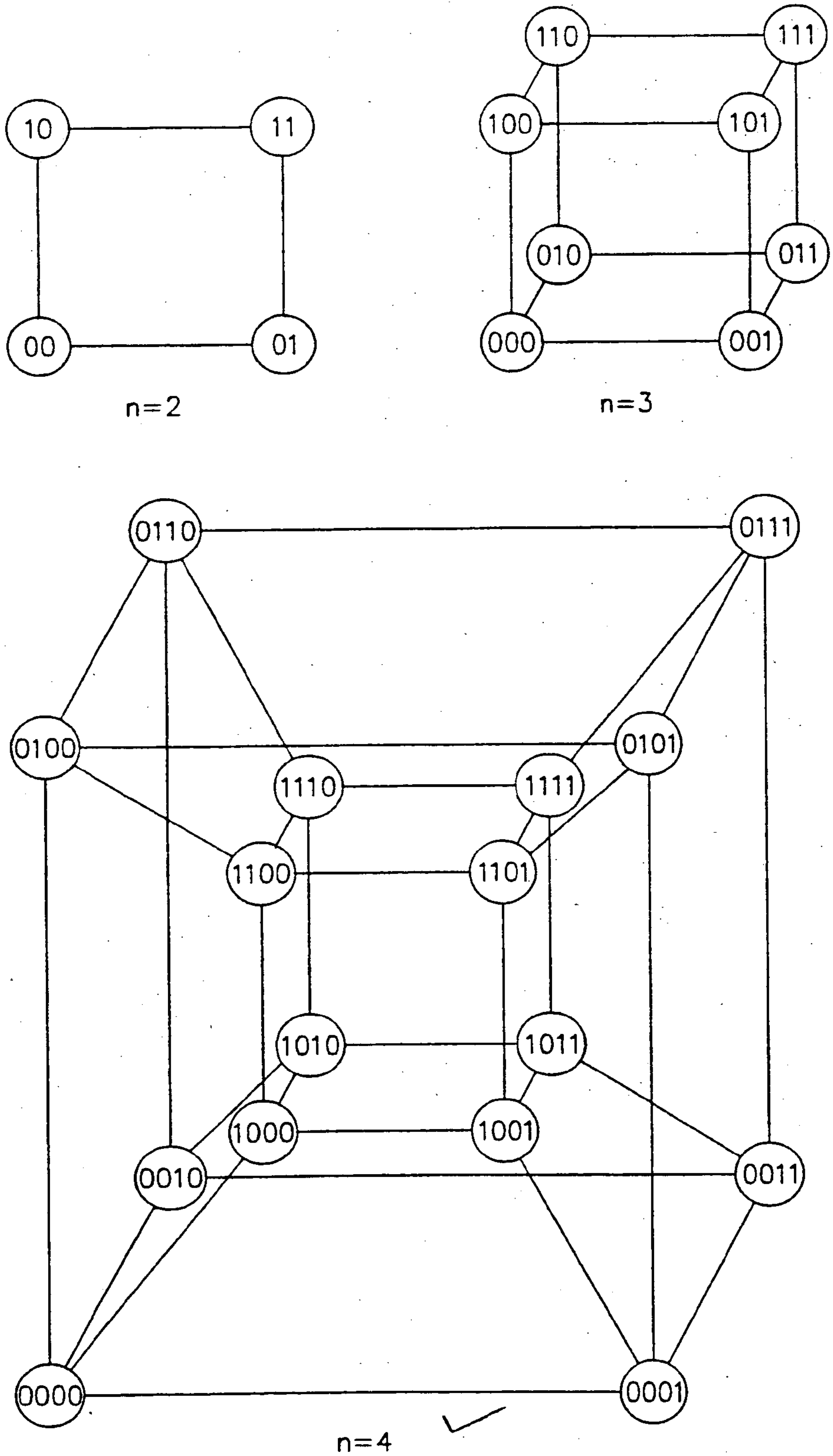
10
11
00
01
n=2
110
111
100
101
010
011
000
001
n=3
0110
0111
0100
0101
1110
1111
1100
1101
1010
1011
1000
1001
0010
0011
0000
0001
n=4

FIG.1

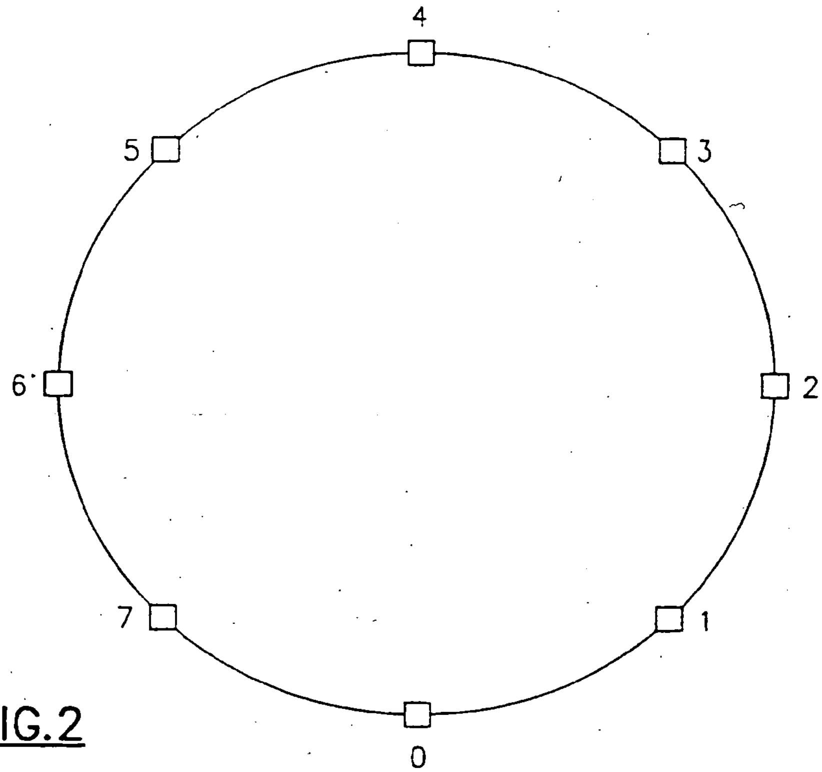

FIG.2

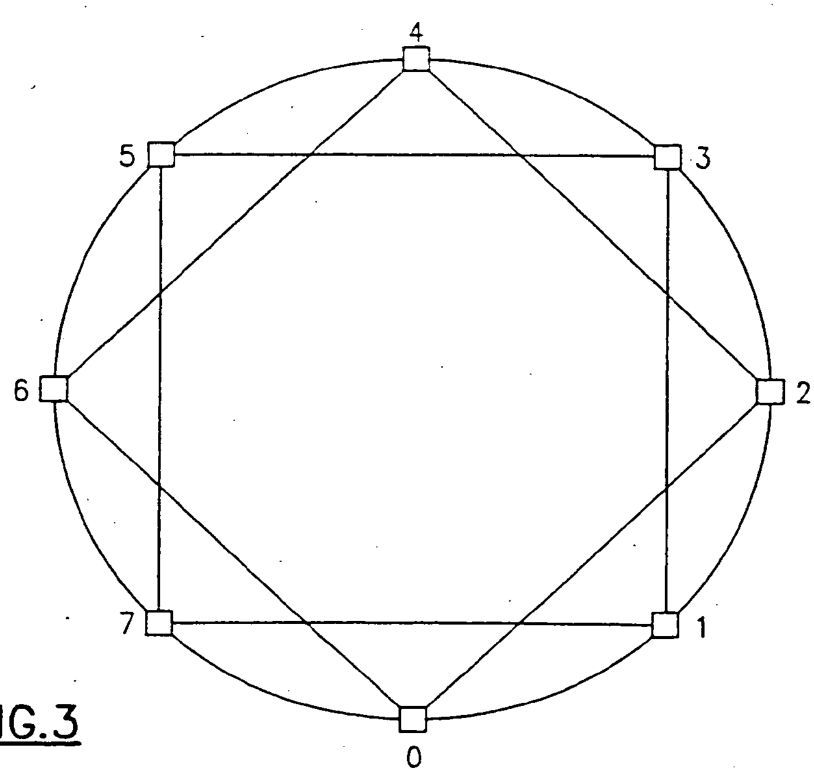

FIG.3

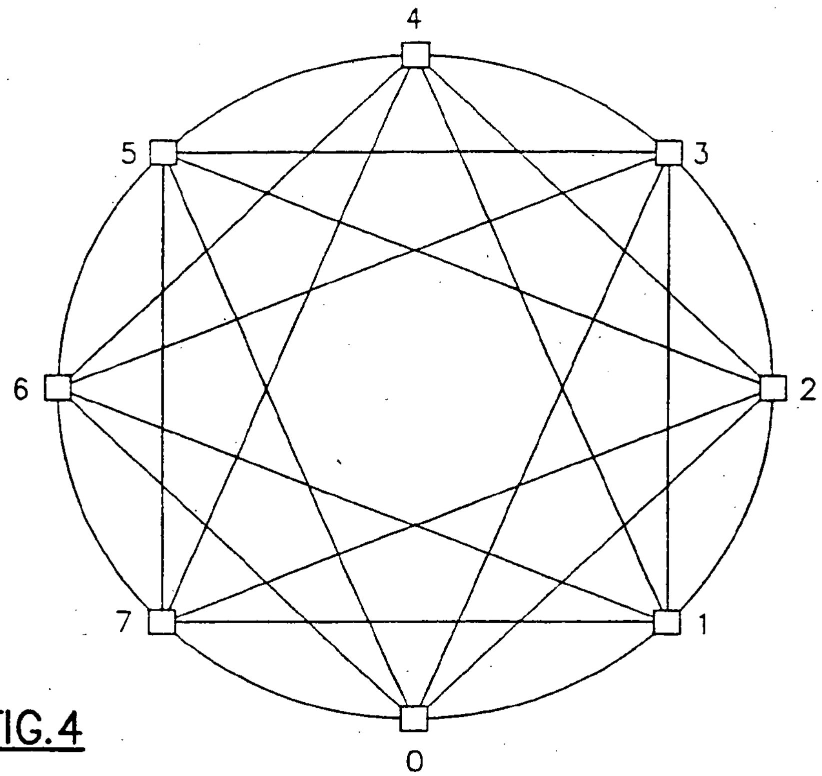

FIG.4

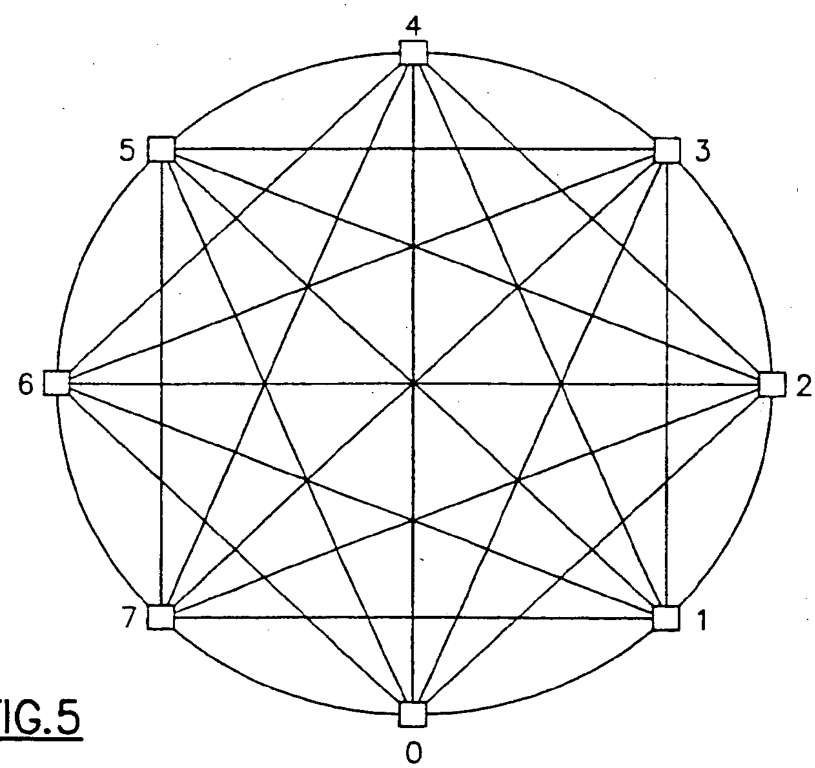

FIG.5

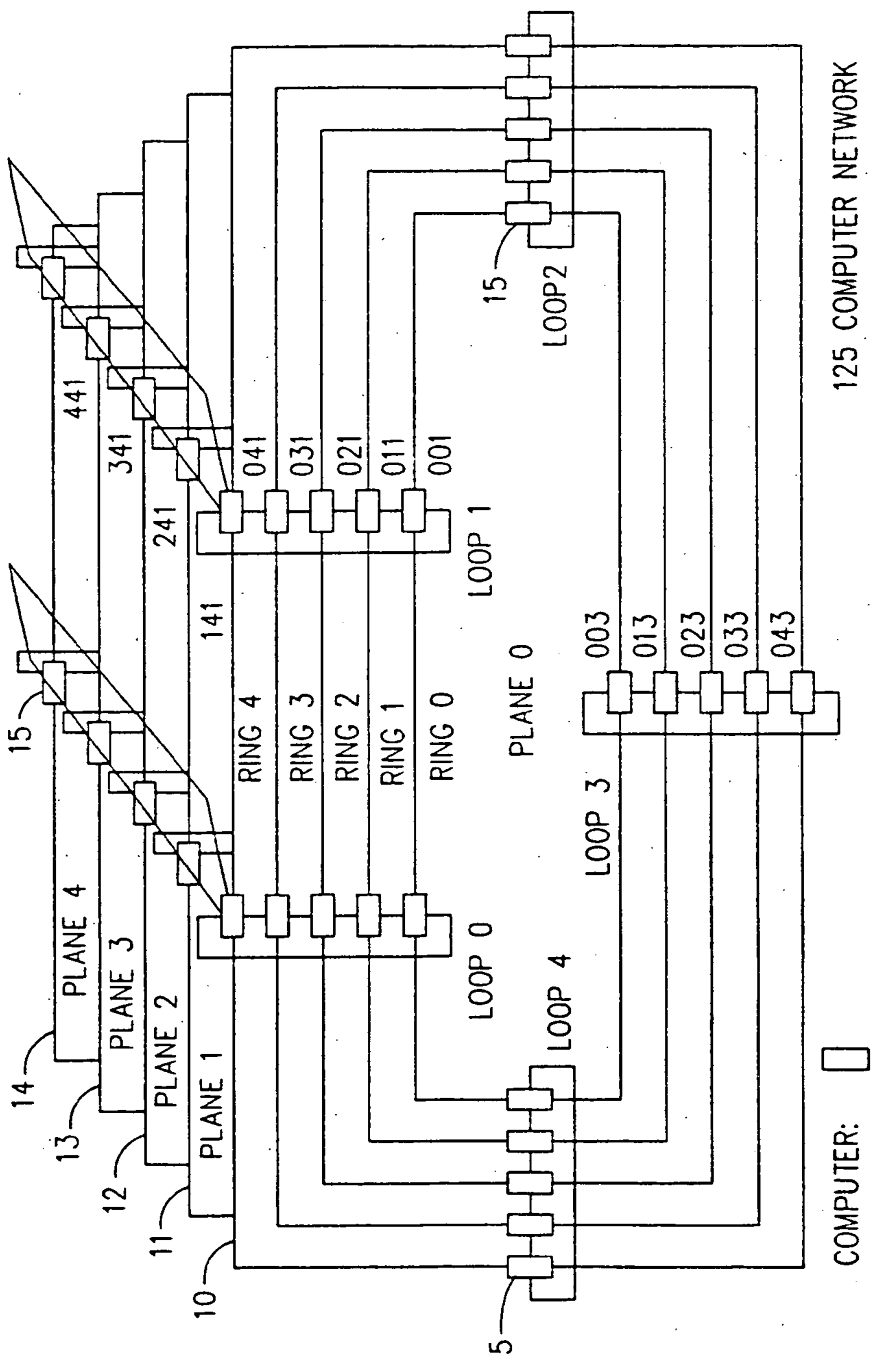
14
13
12
11
10
15
PLANE 4
PLANE 3
PLANE 2
PLANE 1
441
341
241
141
041
031
021
011
001
RING 4
RING 3
RING 2
RING 1
RING 0
PLANE 0
LOOP 0
LOOP 1
LOOP2
LOOP 3
LOOP 4
003
013
023
033
043
COMPUTER:
125 COMPUTER NETWORK

FIG.6

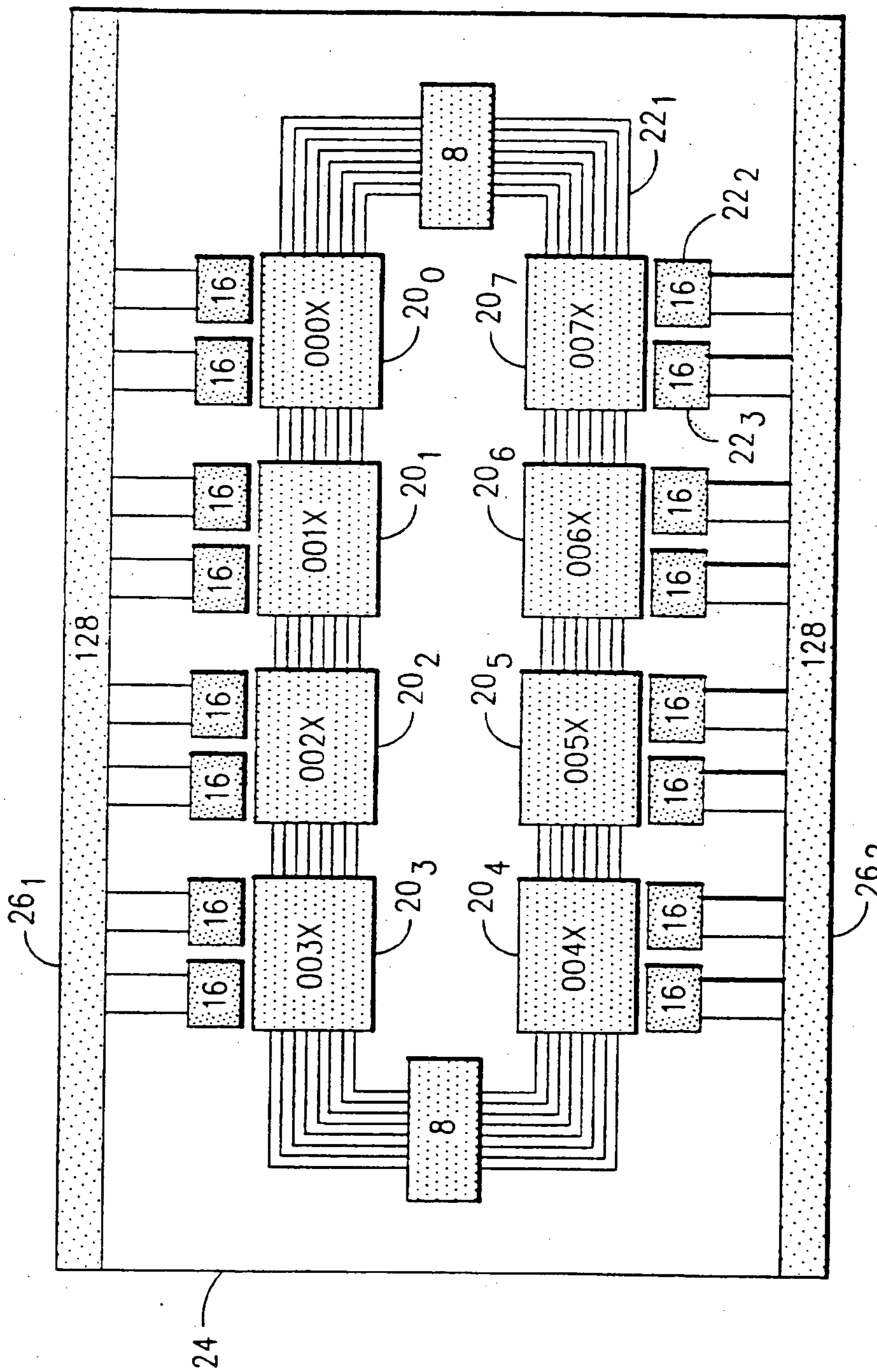
000X
001X
002X
003X
004X
005X
006X
007X
16
8
128
20_0
20_1
20_2
20_3
20_4
20_5
20_6
20_7
22_1
22_2
22_3
26_1
26_2
24

FIG.7

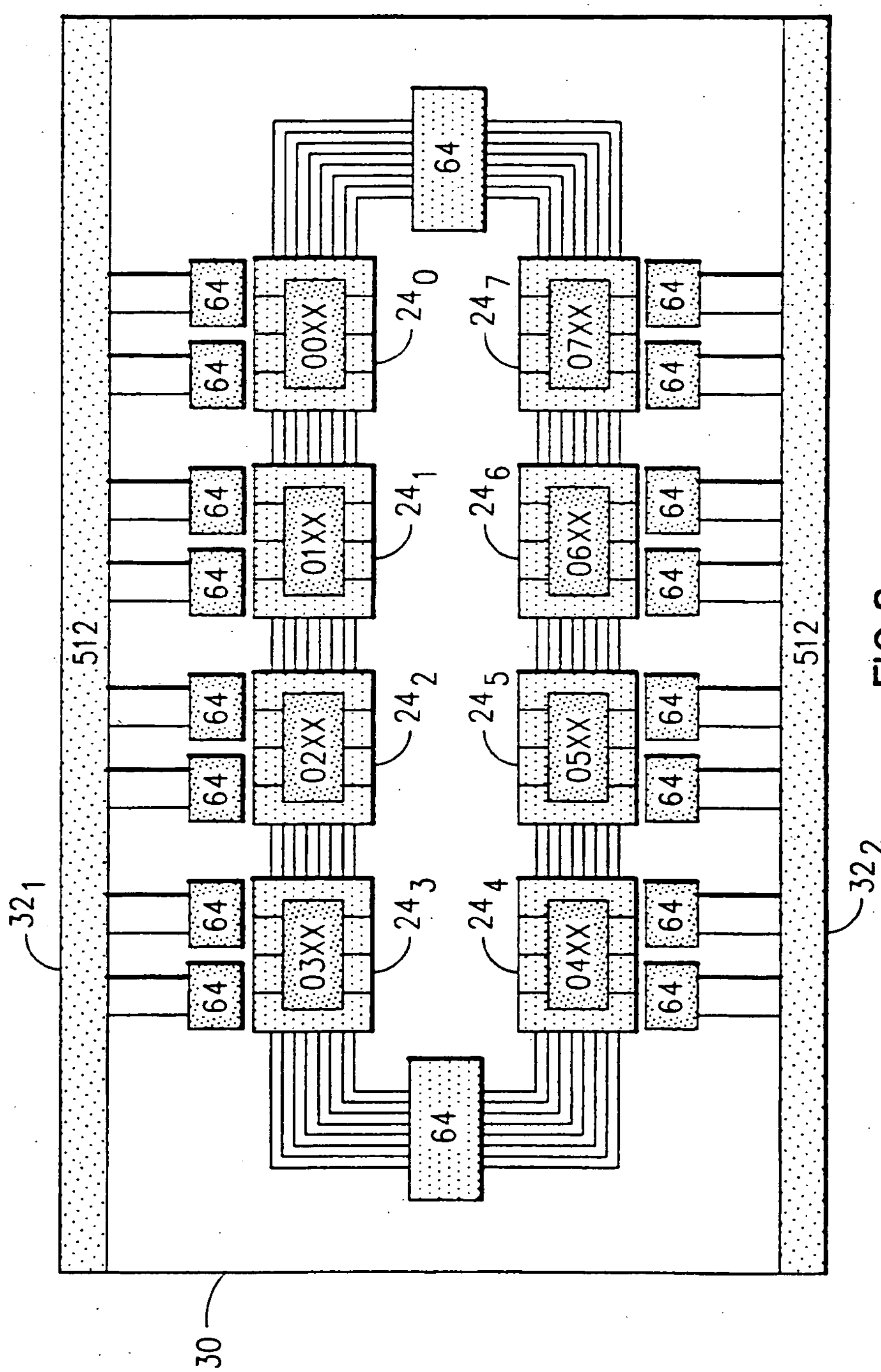
64
00XX
01XX
02XX
03XX
04XX
05XX
06XX
07XX
24 0
24 1
24 2
24 3
24 4
24 5
24 6
24 7
512
32 1
32 2
30

FIG.8

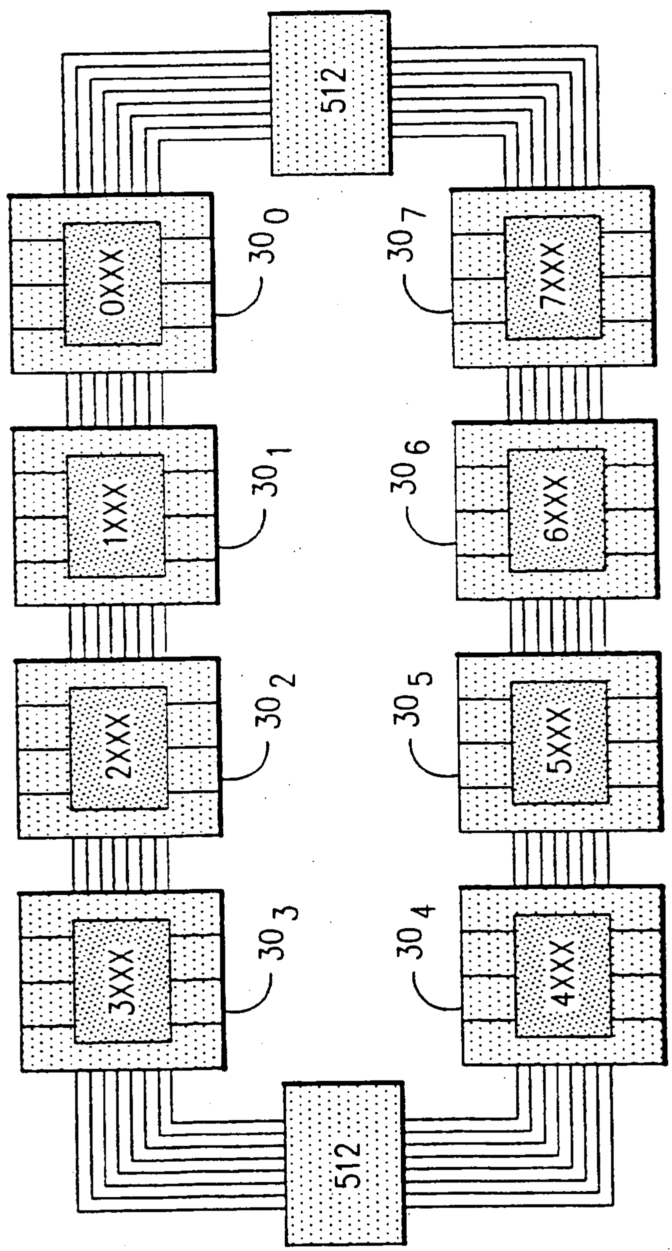
512
0XXX
1XXX
2XXX
3XXX
4XXX
5XXX
6XXX
7XXX
512
30_0
30_1
30_2
30_3
30_4
30_5
30_6
30_7

FIG.9

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 514 043 A3**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **92303900.2**

(51) Int. Cl.5: **G06F 15/16**

(22) Date of filing: **30.04.92**

(30) Priority: **13.05.91 US 698866**

(43) Date of publication of application: **19.11.92 Bulletin 92/47**

(84) Designated Contracting States: **DE FR GB IT**

(88) Date of deferred publication of the search report: **18.05.94 Bulletin 94/20**

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: **Rolfe, David Bruce**
24 Pine Tree Road
West Hurley, New York 12491(US)

(74) Representative: **Burt, Roger James, Dr. et al**
IBM United Kingdom Limited
Intellectual Property Department
Hursley Park
Winchester Hampshire SO21 2JN (GB)

(54) **Computing system.**

(57) Computer elements in a massively parallel computer system are interconnected in such a way that the number of connections per element can be balanced against the network diameter or worst case path length. This is done by creating a topology that maintains topological properties of hypercubes yet improves flexibility by enumerating the nodes of the network in number systems whose base can be varied. Topologies are generated in which nodes are not always connected when their addresses differ in a single digit. A new variable d is introduced, the purpose of which is to control the overall density of the network by controlling the number of intermediate arc connections within the rings of the network.

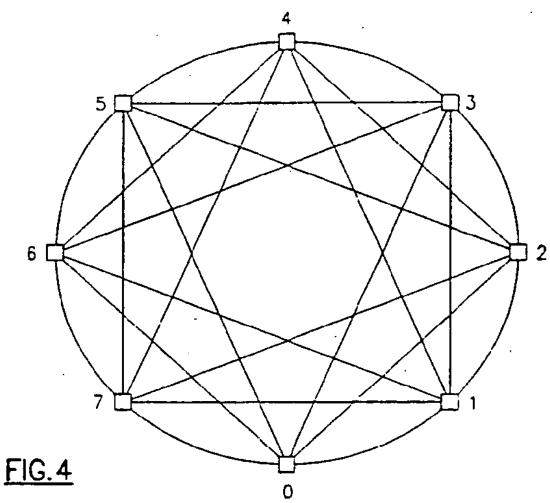

FIG.4

EP 0 514 043 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 92 30 3900

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
D,A	CONFERENCE PROCEEDINGS OF THE 9TH ANNUAL SYMPOSIUM ON COMPUTER ARCHITECTURE 26 April 1982 , AUSTIN, USA pages 90 - 98 L. N. BHUYAN AND D. P. AGRAWAL 'A general class of processor interconnection strategies' * the whole document *	1-8	G06F15/16 G06F15/80
A	THE 9TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS 5 June 1989 , NEWPORT, CA, USA pages 254 - 262 K. EFE 'Programming the Twisted-cube architectures' * the whole document *	1-8	
A	THE 10TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS 28 May 1990 , PARIS, FRANCE pages 262 - 269 N. TZENG 'Structural properties of incomplete hypercube computers' * the whole document *	1-8	TECHNICAL FIELDS SEARCHED (Int.Cl.5) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	9 March 1994	Michel, T

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
: non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document